Video Compression : Wavelet Based Coding and Texture Synthesis Based Coding

Heechan Park and Graham R. Martin
Department of Computer Science
University of Warwick

June, 2005

Abstract

This report introduces the emerging Wavelet-based video coding system. Inefficiency of the block-based motion estimation in context of the wavelet coder is discussed in comparison to the mesh-based approach. A progressive mesh refinement technique using multiple layers is proposed also a fast motion estimation based on the redundant wavelet is presented that outperforms a full-search motion estimation in both computation requirement and motion accuracy. Finally, the two-component texture synthesis algorithm is explored in connection with recently developed technique using GMM and LM to model and estimate transformation. The texton and ICA is briefly explained as alternatives.

Keywords : wavelet, video coding, motion estimation, mesh, texture synthesis

Contents

List of Figures

Chapter 1

Introduction

The success in micro-electronics and computer technology coupled with the creation of networks operating with various channel capacities opened an era of digital communication. The high-speed and reliability of digital communication allowed evolution of communication. For instance, visual communication over networks became a ubiquitous part of our life and the availability and demand for visual communication has already started to outpace increase in channel capacity and storage, thus compression and adaptability to channel capacity are important research areas. In particular, video is a rich and excessive form of media which consists of consecutive images. Even with state-of-art still image compression, video presents a serious storage problem. Undoubtedly transmitting such data over a network is one of most bandwidth-consuming modes of communication.
However, video contains a great deal of redundancy along the time axis. The motion compensation scheme adopted by the video coding standards offers an effective method of inter-frame coding. The block-matching algorithm is a common approach. However, the block-matching technique cannot cope with true motion. In contrast, a mesh-based motion compensation scheme provides effective modeling of complex motion such as rotation, scaling, shearing as well as translation due to warping operation; however it requires relatively high computation. Recently, the mesh-based scheme has been brought back into the light due to an advance in computing power as well as a good fit with the wavelet video coder, which is of current interest in the video coding community. Various channel capacity of end-users' system and the unstable channel condition has proved that a fixed bitrate video compression scheme is not an effective over networks. Demands for a dynamic scheme driven by end-users' systems and channel conditions led to 3D wavelet coding. The multi-resolution paradigm behind the wavelet applies not only spatially but also tempo-

rally. It provides so-called *universal scalability* which allows a flexible decoding mechanism in terms of computational complexity as well as reconstruction quality. Thus, it is possible to control spatial or temporal resolution from the decoder side to suit end-user's system by partial extraction from a full video stream. In the same context, the server is not even required either to compress the same data for each target bitrate or to switch end-users' line to low-resolution version when the channel becomes unstable.
In the long term view, however, video compression seems to have reached its full potential with the conventional frequency based techniques such as the discrete cosine transform. Scalable video coding utilising the wavelet transform can be realised as the scalability function is added to the compression system to meet the requirement of communication over a network as mentioned above. In order to approach higher levels of compression, it is necessary to better understand how human visual perception works, so semantically meaningless information can be removed or synthesized with minimum effort. Very little is known about how the cells in the early visual pathway of primates are grouped into larger structures in higher level as human perceives. Also, it is ambiguous what the generic image representation beyond the image pyramids is. Nevertheless, the compression based on texture synthesis is likely to provide a much more compact representation although extensive research and experiments need to be conducted.

The research carried out throughout this year is in two areas: i.e. wavelet based coding and texture synthesis based coding. Although the main work in this report focuses on the former, we realise that texture synthesis based coding has great potential for future generation low-bitrate coding and much of our work is still in progress, as briefly reviewed at the end of this report. This report is organised as follows.

- Chapter 2 highlights the principle behind wavelet based video coding , how the motion compensation scheme is combined via a lifting scheme and an advanced video motion model:(warping).

- Chapter 3 continues to describe mesh-based motion estimation with a progressive mesh using layered partial refinement, i.e. a method of mesh re-generation without transmitting overhead information. Particularly, we discuss its potentials in a wavelet coder. This work has been reported to VLBV 2005 [21]

- Chapter 4 explores an efficient motion estimation method in the redundant wavelet domain in connection with a multi-hypothesis scheme based on phase diversity of the

redundant wavelet. This work has also been reported to VLBV 2005 [19]

- Chapter 5 introduces texture synthesis and discusses affine transform based texture synthesis. The fundamental approach using the generative texton is briefly mentioned.

- Chapter 6 summrises this report, conclusions are drawn and future research directions suggested. In particular, an extension of texture synthesis to 3D is briefly discussed.

Chapter 2

Scalable Video Coding

2.1 Overview of Inter-frame Wavelet Coder

Wireless technology and mobile devices today, such as the PDA, show the importance of scalable video coding while visual communication is an essential part of our life. In other words, multicast video transmission in a heterogeneous environment of terminals and networks with variable bandwidth requires a flexible adaptation with regard to network properties, terminal capabilities, and user needs. A video coding concept is desired that provides spatial, temporal, and quality scalability and a performance comparable to that of state-of-art non-scalable coders. Scalability is hard to realise in the traditional hybrid coder incorporating motion compensated prediction and block transform coding. This recursive encoding and decoding(*closed-loop*) is to ensure that encoder and decoder work synchronously. If the encoder and decoder do not relate to the same reference in the prediction loop, an error is induced at the decoder and the error will accumulate with time (*drift*). The wavelet transform has proved its efficiency for still image compression like JPEG2000[13]. The multi-resolutional nature of the wavelet inherently provides scalability. Three-dimensional subband decompositions have been reported quite early in the literature [14]. The *open-loop* architecture of temporal motion-compensated wavelet decomposition via a lifting scheme was reported in [6]. This framework has shown that the lifting scheme along the time axis can be combined with an arbitrary motion compensation method, resulting in promising performance. Spatio-temporal subband coding schemes applying a subband decomposition along the temporal axis of a video sequence combined with a motion compensation scheme became an alternative approach to hybrid coding.
Motion compensated temporal decomposition combined with a spatial wavelet transform

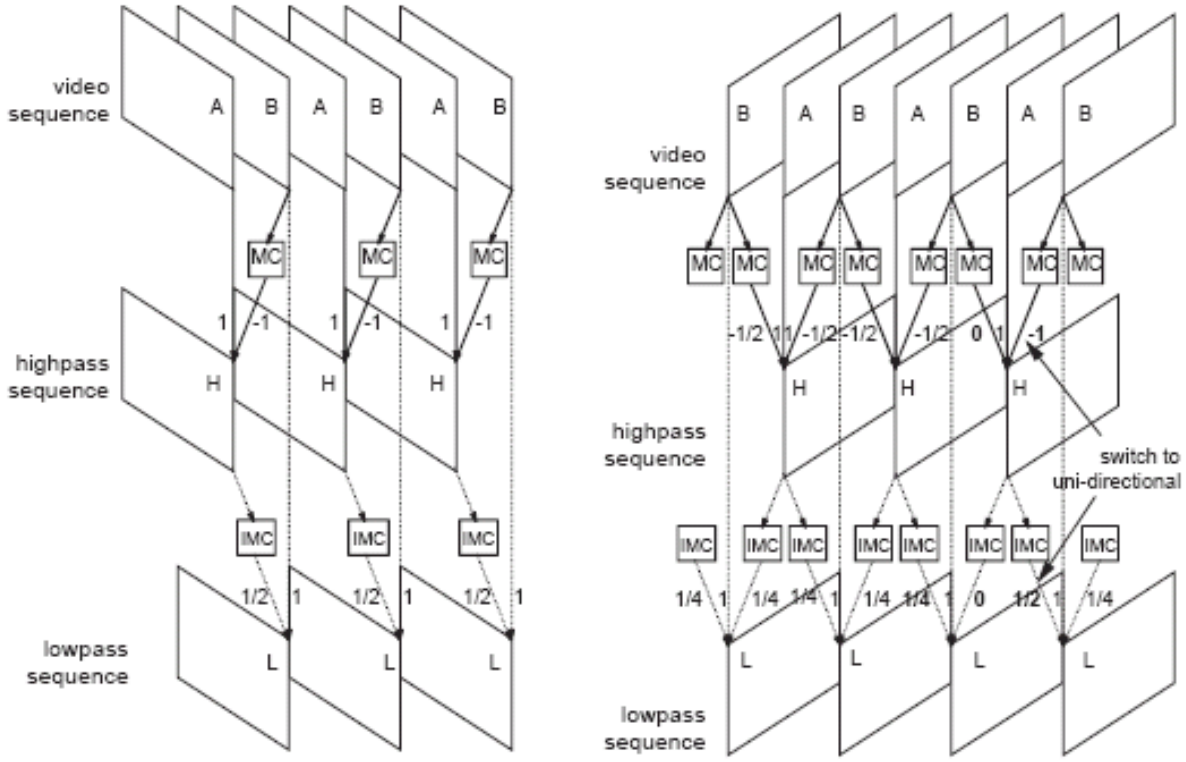


Figure 2.1: (left)Haar filter with uni-directional prediction and update; (right)5/3 filter with bi-directional prediction and update (taken from Ohm *et. al.*[17])

forms a fully scalable inter-frame wavelet coding architecture. There are two possible implementations of the system, t+2D and 2D+t. In the t+2D scheme, temporal decomposition is followed by spatial decomposition, and this has shown much closer performance to that of state-of-art hybrid coder[10]. In the 2d+t case, motion estimation is applied in the wavelet domain which exhibits shift-variance. The overcomplete wavelet transform was proposed to overcome the shift-variance problem[28]. More information on options enabled by the different implementation of the wavelet coding and motion compensation on the overcomplete wavelet domain will be explored in Chap. 4. We shall discover more on motion compensated temporal filtering in the next section.

2.2 Motion Compensated Temporal Filtering

A common implementation of motion compensated temporal decomposition (MCTF) is a combination of the Haar transform and block-based motion compensation. Fig.2.1(left)

shows a Haar-based MCTF with pairs of adjacent input frames A and B and lowpass frames L and highpass frame H. The motion compensated prediction step and update step are defined by

$$H = (A - \hat{B})/\sqrt{2} \tag{2.1}$$

$$L = (\check{A} + B)/\sqrt{2} \tag{2.2}$$

where $\hat{X}$ and $\check{X}$, e.g. A, B represent motion-compensation and inverse motion-compensation respectively. The reconstruction equations are given by

$$B = (L - \check{H})/\sqrt{2} \tag{2.3}$$

$$A = (\hat{L} + H)/\sqrt{2} \tag{2.4}$$

The lifting operations are applied per pixel for uniquely connected pairs of pixels. In the case of local motion, there may be unreferenced regions in frame B, called *unconnected* pixels. The update step for unconnected pixels is given by

$$L = \sqrt{2}B \tag{2.5}$$

$$B = L/\sqrt{2} \tag{2.6}$$

Also, a pixel in frame B may be *multi-connected*. In this case, the prediction step given by Eq.2.1 is applied to any referencing pixel in frame A, while the update step is performed for only one of the referencing pixels as a unique connection. There also may be pixels in frame A, which cannot be properly matched with areas in frame B. These are called *intra-pixels*. The occurrence of different pixel classes is illustrated in Fig.2.2.

An important issue concerning temporal multi-resolution analysis is the choice of the temporal filter length. Long filters take better advantage of the temporal correlation existing between successive frames but have an increased motion overhead, are more complex and have higher latency than Haar filters. The 5/3 temporal transform is a good compromise whose benefits have been presented and justified by Secker and Taubman [24] Fig.2.1(right) shows a 5/3 filter based on the MCTF structure. The corresponding predic-

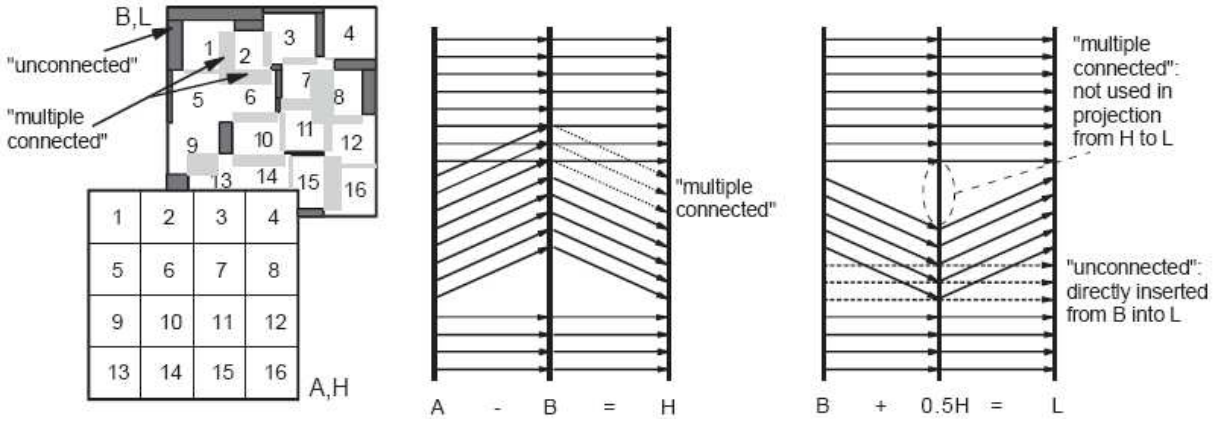


Figure 2.2: Basic decomposition element of Haar-based MCTF in lifting implementation with block-based MC (left) and associated pixel classes (right) (taken from Ohm *et. al.*[17])

tion and update step are given by

$$H = A - \hat{B}/2 - \tilde{B}/2 \tag{2.7}$$

$$L = B + \check{H}/4 + \bar{H}/4 \tag{2.8}$$

where $\hat{X}$, $\tilde{X}$, $\check{X}$ and $\bar{X}$, e.g. A, B represent backward , forward, inverse backward and inverse forward motion compensation. Analogous to Haar-based lifting, a reconstruction is given by

$$B = L - \check{H}/4 - \bar{H}/4 \tag{2.9}$$

$$A = H + \hat{B}/2 + \tilde{B}/2 \tag{2.10}$$

It is well-known that perfect reconstruction is an inherent property of the lifting structure, even if the input frame undergoes a non-linear operation such as motion compensation. However, in order for a lifting structure to exactly implement the transversal wavelet filtering, motion transformation must be invertible. However, as mentioned above, the occurrence of so-call unconnected / multiple connected pixel prevents invertibility and {Eq.2.6,Eq.2.8} do not hold in such cases. In the next section, we discuss mesh-based motion compensation in which a unique trajectory between two frames is guaranteed thus leading to perfect invertibility.

2.3 Advanced Motion Models: Mesh based Motion Compensation

Even with extensive research effort to overcome an intrinsic problem of the block-based motion model, non-invertibility,[10] an efficient block based framework that would successfully compete with state-of-art hybrid coders has not been reported so far. Recently, deformable-mesh motion models have been proposed for wavelet video coding[26]. Applied within a motion-compensated lifting framework these models allow efficient temporal subband decomposition along motion trajectories. Invertibility of the mesh-based motion model overcomes many of the problems observed in block-based motion since the existence of unique trajectories does not allow for existence of the aforementioned disconnected pixels. On the other hand, the composition of motion fields estimated at different levels of temporal decomposition permits a compact representation of motion fields, regardless of the temporal support of the particular transform used. It is important to note that the motion overhead in a mesh-based coder can stay comparable to that of a block-based coder. Fig.2.3 shows warping motion compensation using triangular meshes. The mesh-based motion compensation is based on interpolation of the motion vector field. During motion estimation, the position of each grid point in the previous frame is optimised iteratively while holding the surrounding grid point fixed. The other advantage of using mesh based motion the capability to model more complex motion, i.e. rotation, scaling and translation. Due to the continuity of the interpolated motion vector field, the temporal highpass frames are inherently free of blocking artefacts. A drawback is that the iterative motion estimation process increases the encoder complexity[16]. Often the regular mesh is not flexible enough to cope with fast motion and often produces warping artefacts which correspond to blocking artefacts in a block-based model. This problem can be alleviated by generating a mesh based on the contents of a frame as shown in Fig.2.4. This in turn requires additional bits to re-generate the mesh in the decoder. This problem is dealt with in-depth in the next chapter.

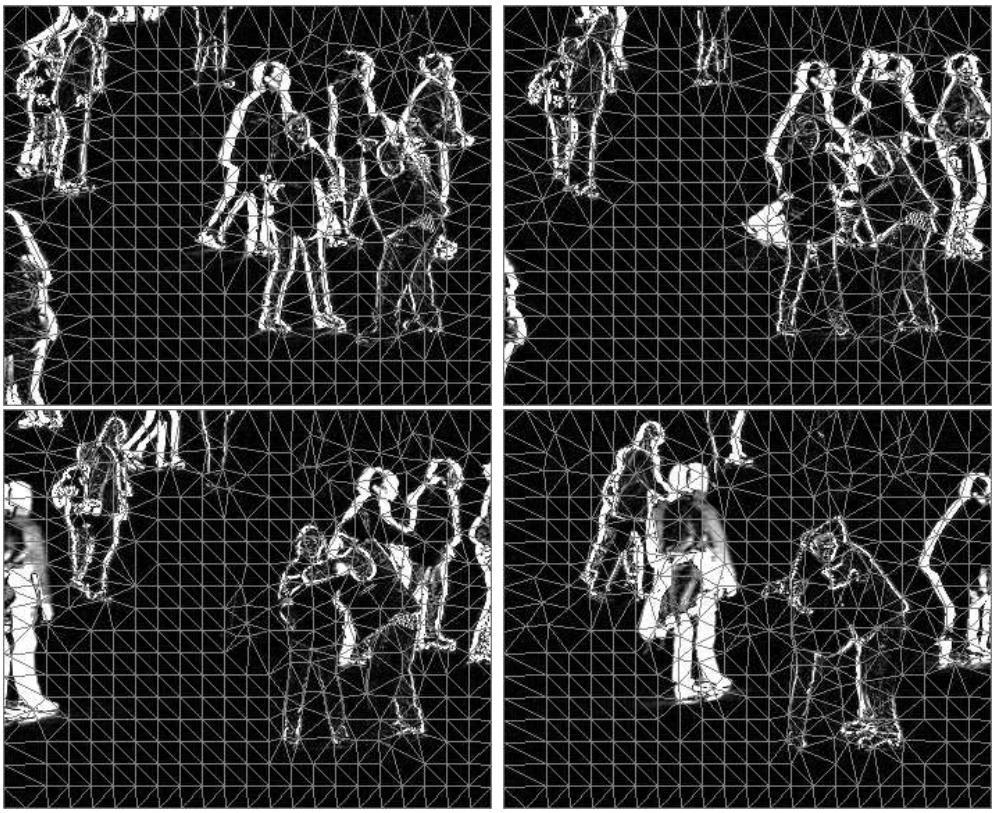


Figure 2.3: Mesh based Motion Compensation: four successive frames with mesh deformation through time

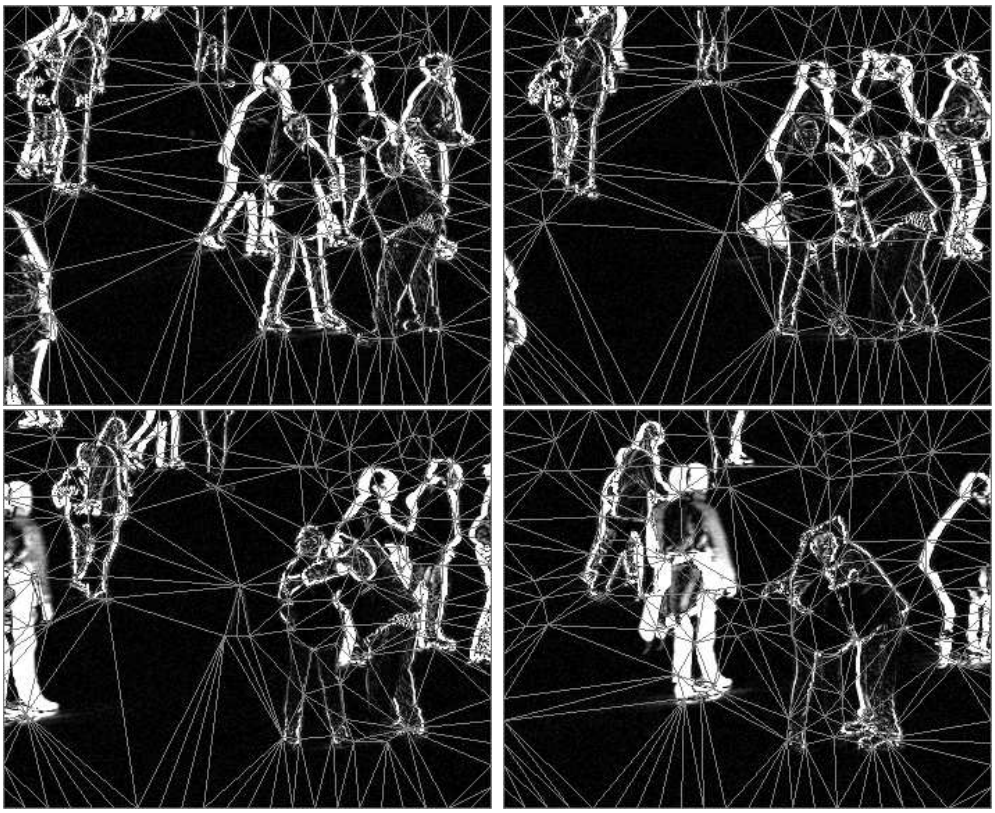


Figure 2.4: Content based Mesh Motion Compensation: four successive frames with content based mesh deformation through time

Chapter 3

Progressive Mesh Refinement

3.1 Introduction

Block matching motion estimation forms an essential component of inter-frame coding in many video coding standards. The block matching algorithm adopts a translational motion model, but this is intrinsically limited when representing real world motion that involves not only translational but also scaling and rotational motion. In order to cope with complex motion such as rotation and zooming, deformable mesh-based algorithms have been proposed. In general, a mesh consists of polygonal patches, and a spatial transformation function is applied to map the image into a new coordinate system.

Mesh-based motion estimation can be divided into two categories, defined by whether motion is estimated in the forward or backward directions. In backward motion estimation, a mesh is applied to the current frame and deformations are estimated from the current to the reference frame. Forward methods operate in the opposite manner. Backward motion estimation is widely used because of its relative simplicity and the lower computational requirement of the mapping process. Forward methods can provide adaptive deformation of the patch structure to track feature changes in the image, but at the expense of complexity. Mesh-based techniques can be further categorised depending on whether regular or irregular mesh is employed. A regular mesh consists of uniform patches. An irregular mesh is generated according to the image content using Delaunay or Quadtree methods, and where patch size varies with intensity gradient or motion activity. Generally, a regular mesh is associated with backward estimation. The irregular mesh is coupled with forward estimation to avoid transmitting a large overhead for the patch structure. The latter provides better performance than a regular mesh. However it is not popular in real

applications due to the high complexity of the forward method or associated overhead for transmitting node positions. In this paper, we present a regular mesh technique with backward estimation that features the advantages of an irregular mesh.

3.2 ME / MC with Spatial Transform

Motion estimation (ME) and compensation (MC) based on a triangular mesh, partitions the image into a number of triangular patches where the vertices are denoted as grid points. Using mesh refinement (Section 2.2), the displacement of each grid point is estimated and represented by a motion vector(MV). The displaced grid points define a deformed mesh that describes the underlying motion. The deformed mesh of the reference frame is obtained from estimating the displaced position of the mesh vertices in the current frame. Motion compensation proceeds by retrieving six affine parameters from the displacements of the three vertices of each triangular patch, and synthesizing patch content using a warping operation defined by the six parameters.

3.2.1 Affine Transform

An affine transform models translation, rotation, and scaling of a patch in the current frame to the corresponding distorted patch in the reference frame. This transformation is represented by six parameters. An intensity value of pixel (x, y) in the i th synthesized patch $\hat{P}$ in the predicted frame K is given by

$$\hat{P_i^k}(x, y) = P_i^{k-1}(f_i(x, y)) \tag{3.1}$$

where the affine transform $f(\cdot)$ of the patch is given by

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.2}$$

where (x', y') and (x, y) denote positions in the reference frame and the current frame respectively. There is a one-to-one correspondence between vertices in the current frame and the reference frame, and therefore, the six parameters , $a_1, a_2, a_3, b_1, b_2, b_3$ are obtained by solving equations provided by the motion vectors at the vertices and pixels within corresponding patches can be interpolated accordingly.

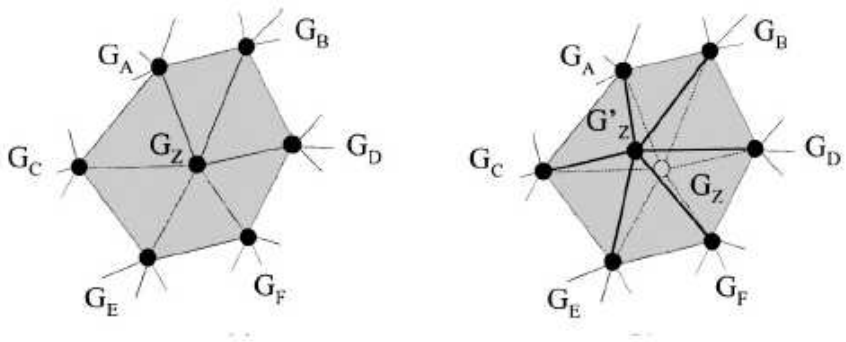


Figure 3.1: Mesh refinement of the hexagonal matching, (left) before refinement, (right) after refinement

3.2.2 Mesh Refinement

Mesh refinement refers to modification of the grid point locations so that the image intensity distribution within any two corresponding patches in the current and reference frame match under an affine transformation. Finding the optimum combination of each grid point that minimizes the difference between the current and reference frame for all possible combinations of grid point locations is not feasible in practice. Instead, finding a sub-optimum combination by successively visiting each grid point of the mesh and moving the grid point to a new position within range, thus preserving mesh connectivity and minimizing matching error locally has been developed by Nakaya *et al.* [16].The ME of the grid points proceeds with iterative local minimisation of the prediction error to refine the MV as below.

While keeping the location of the six surrounding grid points, $G_A \sim G_F$ fixed(Fig.3.1), G_Z is moved to an adjacent position G'_Z. This is repeated, and for each move, the six surrounding patches inside the hexagon are warped and compared with the current patch by the mean absolute difference. The optimum position of G'_Z is registered as the new position of G_Z. This procedure is repeated for each grid point until all the grid points converge to either local or global minima.

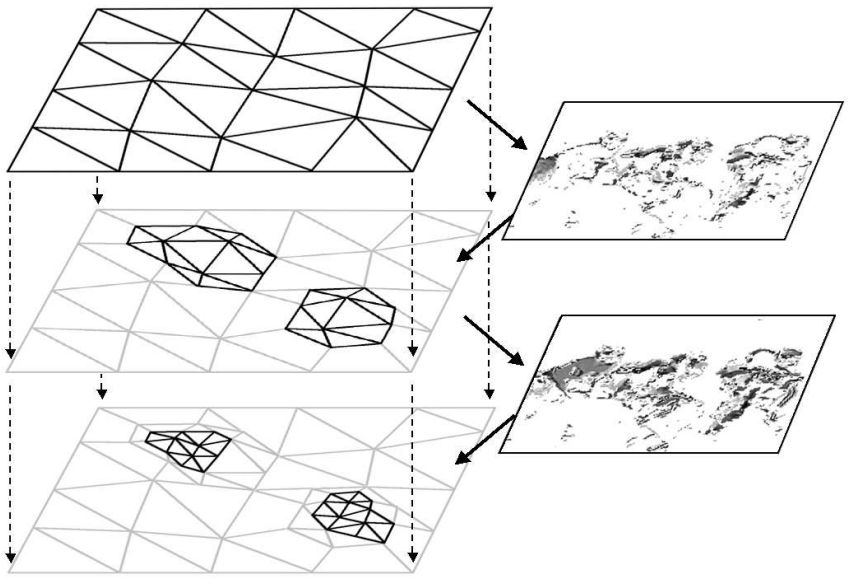


Figure 3.2: Progressive mesh refinement layers and approximated frame difference maps

3.3 Progressive Motion Estimation

A block-based model leads to severe block distortions while the mesh-based method may cause warping artifacts. In terms of prediction accuracy, the mesh-based model can give a more visually acceptable prediction, particularly in the presence of non-translational motion. The complex motion modelling and block artifact-free characteristic of warping enables identification of the approximate difference region between successive frames using the motion vectors alone. This does not appear to have been explored, and is the motivation behind our proposed algorithm.

As mentioned, regular / backward mesh ME is a popular choice due to its relative simplicity and lower computational cost. However, a uniform patch structure cannot accommodate non-stationary image characteristics nor cope with exact object boundary representation. This is addressed by employing a hierarchical structure using a Quadtree[12]. A mesh of a different density is applied according to motion activity, and this allows more accurate motion modelling where needed. However, the technique requires additional bits to indicate the motion active region and has constraints on the flexible movement of the grid points, which add complexity to the refinement process. We propose a progressive mesh refinement method that is adaptable to motion activity in a layered fashion, which overcomes the limitation above.

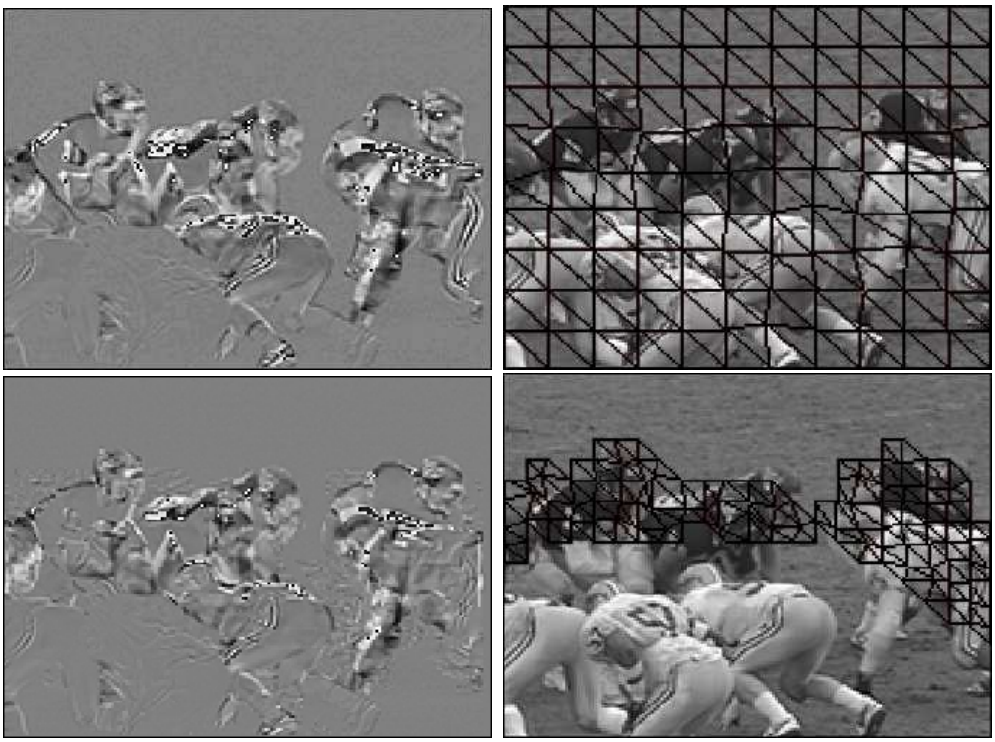


Figure 3.3: Frame difference and partial refinement (from top to bottom and left to right)(a) difference between current and reference images, (b) deformed regular mesh, (c) difference between frame synthesized by (b) and current frame, (d) partial refinement

Fig.3.2 illustrates progressive motion estimation in terms of layer and an approximated frame difference between layers to identify motion-active regions and generate the partial mesh of the next layer. A global motion estimation is performed on the top layer while the next layers concentrate on the finer motion activity. A similar approach was suggested in [27]. While our technique provides partial refinement by employing a layered mesh topology without overhead, Toklu et. al. focus on the hierarchical mesh structure to the entire image in one layer, resulting in higher bit rates. Fig.3.3 shows an example of an approximated difference map. The actual frame difference between the current and reference frames,(a), is quite well approximated in (c). The approximate difference, (c), is obtained from subtracting the (warped) predicted reference frame (b) from the current frame.

The technique proceeds as follows. Firstly we apply mesh refinement for a coarse prediction with a uniform mesh and synthesize an image from the resulting MVs as in the standard mesh-based algorithm. We then identify regions where motion discontinuities

exist from the difference map, that is the variance of the patch difference,v_P is greater than the variance of the frame difference, v_F. The variance of the frame difference is given by

$$v_F = \frac{1}{M \cdot N} \sum_{j=1}^{M} \sum_{i=1}^{N} (f(i,j) - \bar{f})^2 \tag{3.3}$$

where M and N are the frame dimensions and $\bar{f}$ denotes the mean frame difference. The variance of the patch difference is given by

$$v_P = \frac{1}{K} \sum_{i=1}^{K} (f_P(i) - \bar{f_P})^2 \tag{3.4}$$

where K refers to the number of pixels in the patch.

In the next step, a finer regular mesh is applied to the regions containing motion discontinuities, as depicted in Fig.3 (d). Consequently we have layers of mesh, a coarse mesh that covers the entire image and denser partial mesh applied to the moving regions only. This allows a hierarchical refinement without the explicit overhead and constraints on movement of the grid points.

3.4 Experiments

The algorithms were evaluated using the QCIF resolution test sequences "Crew", "Football", "Ice", "Suzie" and "Stefan". According to our experiments, using two layers of mesh with a patch size of 16×16 and 8×8 show the best performance in QCIF resolution in terms of rate-distortion. The hexagonal matching algorithm[16] is applied with a search range of ± 7 pixels for the first layer and ± 3 pixels for the second layer which preserves mesh connectivity. More grid points in the intial layer do not necessarily lead to either better motion active region identification or a better reconstruction quality when bitrate is considered. This is due to the grid point connectivity constraint that prevents effective estimation when an over-dense mesh covers occluded / discovered areas, and of course the increased number of motion vectors. In this sense, the content-based mesh provides an advantage(see Section 3.5). The motion field is initialized with zero-motion and iteration starts with the grid point closest to the image center. A hexagon that contains at least one patch overlapping with the identified regions is included in the partial refinement process.

	One Layer(HMA)		Two Layer	
Sequence	MVs	PSNR	MVs	PSNR
Crew	532	30.93	1190	30.95
Football	645	22.21	1149	22.32
Ice	415	27.11	913	27.68
Susie	349	37.93	719	38.15

Table 3.1: Experimental Result: bitrate for MV and PSNR (0.2 bpp)

Note there is no need to transmit the region information as the region can be identified using the MVs transmitted in each layer. Motion vectors are differentially encoded using Exp-Golomb. Wavelet coding is applied to the displaced frame residue. In Fig.3.4, the left column shows the performace of the single-layered mesh refinement and the right column represents the performance with an additional layer. The overall performance is improved in all test sequences at a fixed bitrate (0.2 bpp). Frame to frame variations do not only comprise the movement of objects. The poor improvement in the Crew sequence can be accounted for by frames containing a flashing light which is more efficiently compressed with residue coding.

3.5 Future Work

Scalable video coding utilizing the wavelet transform applied to both the spatial and temporal domains (3D-DWT) is of current interest in video coding[17]. The mesh-based ME/MC scheme exhibits several merits when deployed in the wavelet coder[25]. The mesh-based estimation can provide scalable coding naturally by controlling the number of grid points or controlling the number of layers in our algorithm. Also, the unique trajectory of each pixel in a mesh-based motion model overcomes the appearance of so-called "multiple/unconnected" pixels occuring in areas not conforming to the rigid translational model. S. Cui et. al. introduced a content-based mesh based on a redundant wavelet[7]. However, it is non-trivial to retrieve the same content-based mesh generated in the decoder when decoding without overhead, which makes deployment in the wavelet coder prohibitive. Our algorithm generates the partial mesh based on the difference map of predictions from the previous layers. The first layer is initialized with a regular mesh, so the same mesh topology can be re-generated without overhead.

Secondly, an effective trade-off between motion coding and residual coding is of prime im-

portance as indicated by the 'Crew' sequence. The layered mesh provides efficient control of the trade-off. Furthermore, intensity control can be introduced using the existing mesh. Each grid point has an additional parameter for intensity scaling by which pixels inside the patch are interpolated.
Lastly, mesh-based coding is also an efficient model for very low bitrate coding with the advantages as mentioned. Adequate subsampling of each layer of mesh leading to a pyramid structure can provide additional improvement in bitrate for the coding of motion information.

3.6 Conclusion

We have described a simple yet effective algorithm that uses the frame difference generated from a mesh-based motion compensated image to identify regions of motion discontinuity. Motion estimation in these regions is refined using a finer mesh structure. It is notable that the proposed approach provides hierarchical refinement without additional overhead, and with no constraint on the movement of grid point positions. The algorithm can be combined with any regular mesh topology. This work shows an improvement over single-layered mesh refinement technique and also demonstrates an effective way to deploy a 'hierarchial mesh without overhead' in scalable video coding.

Chapter 4

Fast Motion Estimation on Redundant Wavelet

4.1 Introduction

Conventional video compression is based on the motion compensated prediction from reconstructed frames in the decoder in a recursive fashion and residue coding by a frequency based technique such as the discrete cosine transform. Due to the recursion and the inherent drift, scalability is difficult to establish in this so-called hybrid coder. Inter-frame wavelet coding overcomes this limitation by replacing the prediction along the time axis by a wavelet filter[17]. Depending on the order of wavelet filters, wavelet coding systems are defined with different characteristics. We focus on the 2D+t system where a spatial filter is followed by a temporal filter. This system provides a more flexible and adaptive system; for instance, it allows a different level of temporal filtering depending on the spatial subband. However, motion estimation must be performed in the wavelet domain that exhibits shift-variance. There are a couple of solutions that overcome the shift-variance of the wavelet. Kim and Park[20] suggested that a low-band-shift method that uses the wavelet macroblock from the redundant wavelet representation. S. Cui *et. al.*[8] proposed a similar but direct and simple method performed in the redundant wavelet domain. This enables a multi-hypothesis paradigm using phase diversity on wavelet domain, which in turn offers substantial improvement over Kim and Park's approach. Both methods are computationally demanding compared to the spatial motion compensation scheme. We focus on S. Cui's algorithm and attempt to reduce computational complexity.

4.2 Inter-frame wavelet coder

The objective of scalable video coding is to introduce an effective decoding mechanism to suit different end-users. Ideally, the new video compression architecture should provide scalability in all aspects, that is spatially, temporally, and in quality and complexity. The multi-resolution paradigm behind the wavelet transform that can be applied not only spatially but also temporally provides an important approach. However, there is an issue regarding the order of wavelet filters for different dimensions. Although the order of spatial and temporal filters appear insignificant, they provide an important insight into the wavelet video coder and the different properties are observed. We briefly mention a couple of options enabled by the 2D+t scheme. Firstly, the accuracy of the motion estimation and GOF per subband can be adaptively determined while this is not viable in t+2D. Secondly, different temporal filters can be utilized at each subband. For instance, bi-directional temporal filtering can be used for the low bands, while only forward temporal filtering can be used for the higher bands. The flexible choice of temporal filtering options makes the 2D+t framework deviate from the strict decomposition scheme as performed in the t+2D to provide a flexible 3D decomposition scheme as shown in Fig.4.1. Finally, motion estimation/compensation combined with the redundant wavelet, whic we shall explore in the next section, provides better PSNR performance than if performed in the spatial domain.

4.3 Redundant Wavelet and Multi-hypothesis

In essence, the redundant discrete wavelet transform removes the downsampling procedure from the critically sampled discrete wavelet transform (DWT) to produce an overcomplete representation. The well-known shift variance of the DWT arises from its use of downsampling; while the RDWT is shift invariant since the spatial sampling rate is fixed across scale. Fig.4.2 illustrates the redundant wavelet decomposition (3 levels). The most direct implementation of the algorithms is a trous; resulting in subbands that are exactly the same size as the original signal. Due to its fixed sampling rate, each RDWT coefficient across scale offers a spatially coherent representation, which in turn guarantees shift invariance. With appropriate subsampling of each subband of RDWT, one can produce exactly the same coefficients as a critically sampled DWT applied to the same input signal. There are several methods to invert the RDWT. The single-phase inverse consists of subsampling the RDWT coefficients to extract one critically sampled DWT from the RDWT and

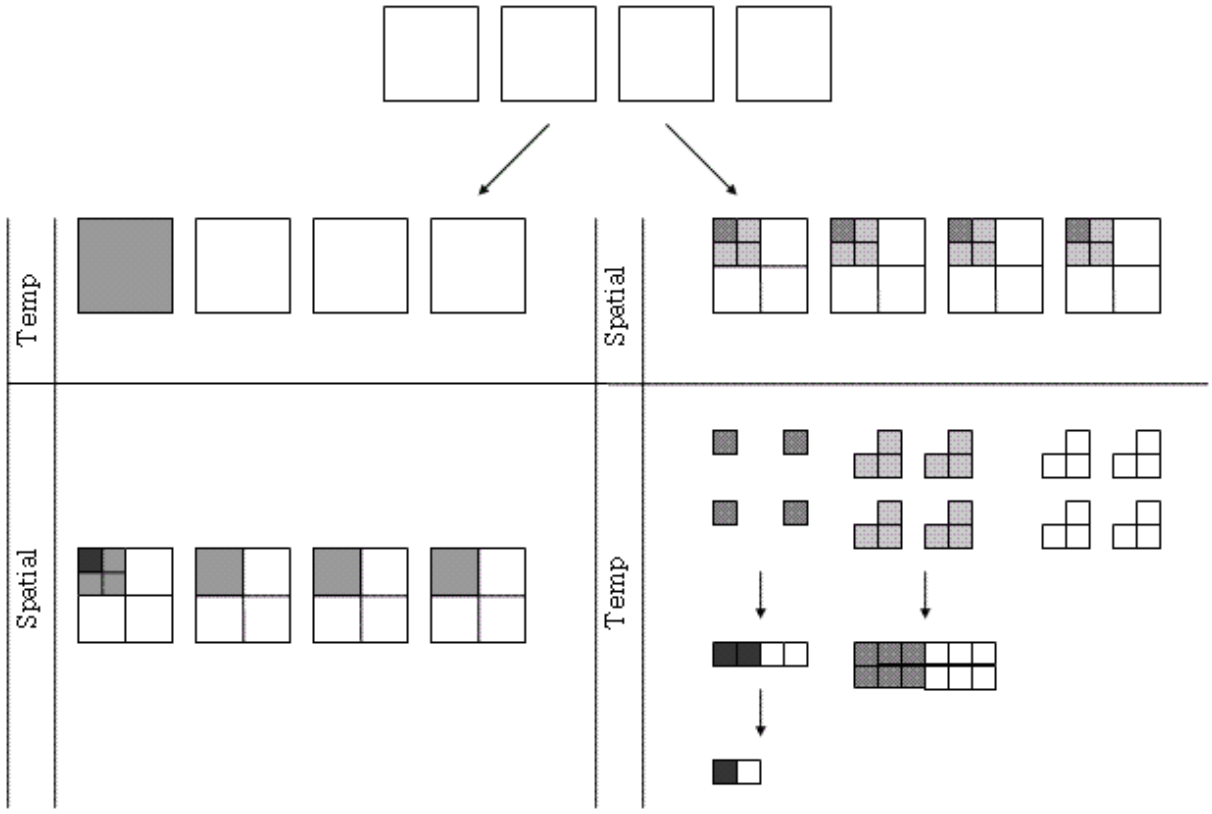


Figure 4.1: 3D Wavelet scheme t+2D(left) and 2D+t(right) architecture

application of the corresponding inverse DWT,

$$Recon[x, y] = InvDWT(sub(RDWT[x, y])) \tag{4.1}$$

where *InvDWT()* is the inverse DWT and *sub()* denotes subsampling to a single-phase. Alternatively, one can employ a multiple-phase inverse, denoted as

$$Recon[x, y] = InvRDWT(RDWT[x, y]) \tag{4.2}$$

In the multiple-phase inverse, one independently inverts each of the 4J critically sampled DWTs constituting the J-scale 2D RDWT and average the resulting reconstructions, or one can employ an equivalent but more computationally efficient filtering implementation. In either case, the single-phase inverse is generally not the same as the multiple-phase inverse. The multiple-phase inverse is shift invariant under linear fractional-pixel interpolation,

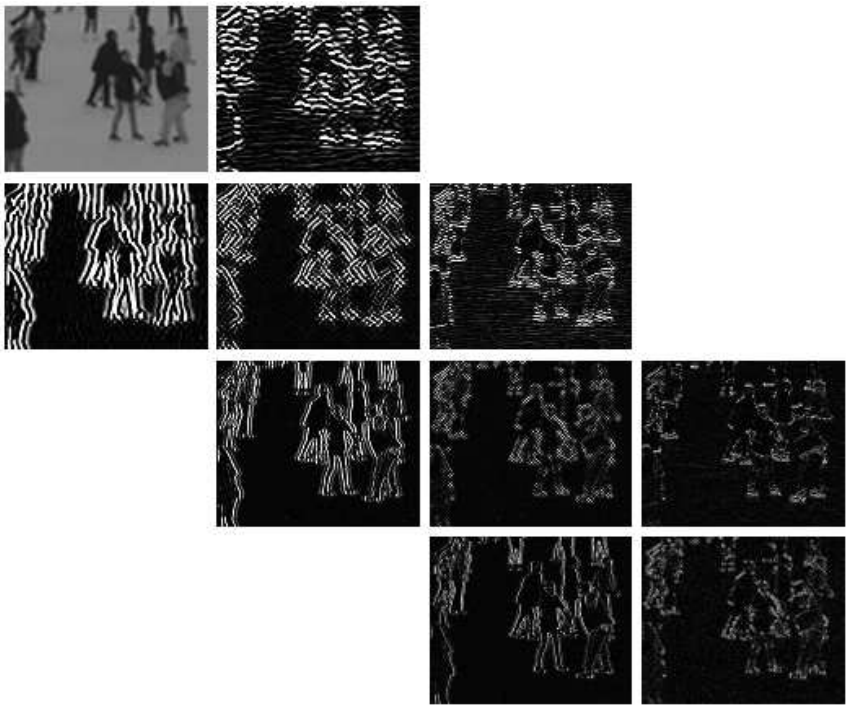


Figure 4.2: Redundant wavelet decomposition

while the single-phase inverse is not. S. Cui. *et. al*[8] extended the above-mentioned multiple-phase inverse operation to a multi-hypothesis paradigm that takes advantage of the diversity of phase in the redundant wavelet representation (RWMH). This system achieves a substantial gain over the single-phase prediction system(RWSH)[20].

4.4 Fast Motion Search

A redundant wavelet representation shows merits for deployment in a scalable video coder, as discussed. However, motion estimation in the redundant wavelet domain increases computational complexity because it requires calculating the absolute difference of blocks across all subband for each position while the number of subbands increases as the level of decomposition. For instance, three levels of decomposition produces ten subbands. As information in the subbands is redundant, it might not be necessary to consider all subbands when calculating the absolute difference. Also, the wavelet representation possesses useful properties such as a hierarchical structure and decomposition into horizontal, vertical and diagonal subbands. These are the motivation behind our investigation for a fast motion

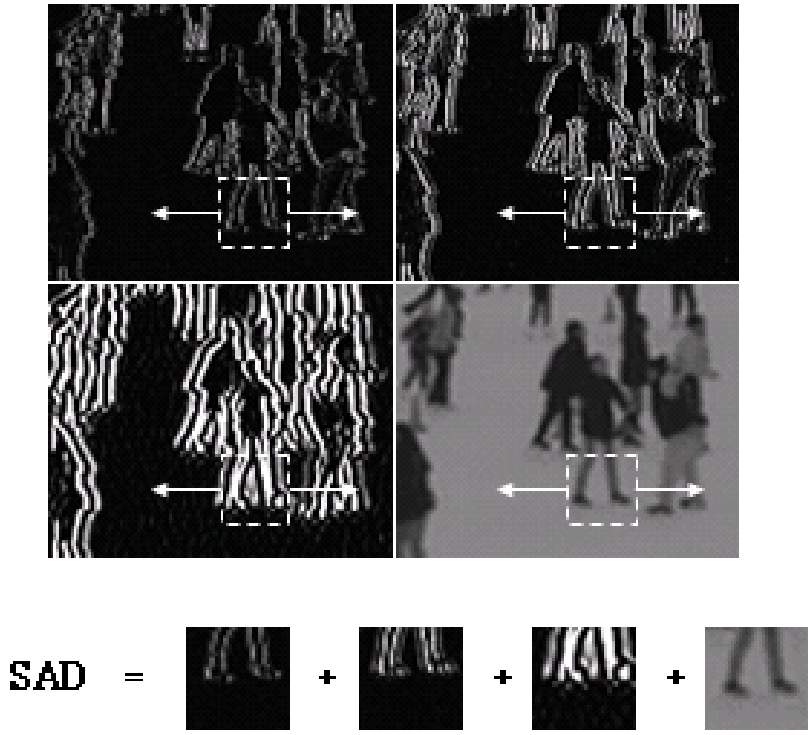


Figure 4.3: LLL and vertical high subbands perpendicular to search direction

estimation technique.

We propose a fast motion search method combining the three-step search and a subband-based search. The three-step search is an effective and fast motion search algorithm[15]. Computation can be further enhanced by selectively considering subbands that are perpendicular to the search direction. For example, for horizontal motion search (x component), the vertical high bands are likely to contribute more than other subbands; thus taking more vertical subbands into account while reducing other subbands as shown in Fig. 4.3.

Our algorithm shares the same overall procedure as the three-step search that determines the optimal search point among nodes of a lattice and repositions a half-sized lattice about the optimal point found. This process is repeated until it reaches 3×3 lattice size. In our algorithm, the full-search methods in each level is replaced by the subband-based search. Fig.4.4 illustrates the subband-based search. The algorithm starts by making two diagonal search point groups and determines an optimum point for each diagonal group. There are two possible scenarios onwards. One includes the center point as optimum in either diagonal group and the other does not. As illustrated in Fig.4.4, the top flow shows

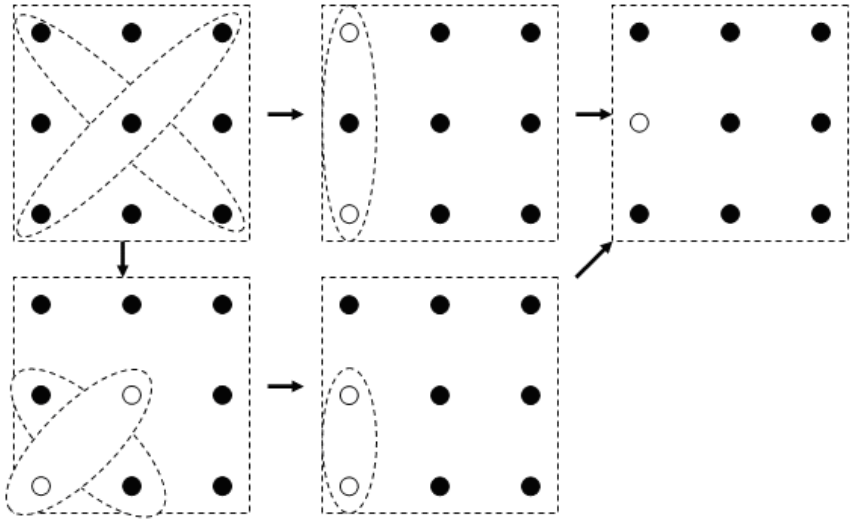


Figure 4.4: Proposed search: two possible scenarios after diagonal search

	RWSH	RWMH	HEIR	1-LEVEL	SPATIAL	TSS	DS	PROPOSED
Ice	39.55	**40.29**	40.28	40.05	**38.94**	39.87	40.15	**39.77**
Crew	37.71	**37.81**	37.80	37.79	**36.62**	37.82	37.79	**37.73**
Football	27.82	**27.89**	27.86	27.85	**27.11**	27.72	27.75	**27.72**
Susie	39.15	**39.79**	39.79	39.79	**39.28**	39.79	39.79	**39.79**
Stefan	26.83	**27.33**	27.32	27.28	**26.81**	27.18	27.30	**27.06**
Complexity	>>	**100%**	51%	44%	**11%**	5%	5%	**2%**
					100%	42%	40%	**19%**
						100%	95%	**44%**

Table 4.1: PSNR results at a fixed bitrate of 0.5 bpp

the latter case and the bottom flow shows the former case. For the latter case, the search finishes by comparing the two optimal points and a point in-between. For the former case, a similar diagonal search is performed again with two optimal points found and two additional points, and finally a comparison between the local optimum point from each diagonal search group determines the final optimum point. Note that the horizontal, vertical and diagonal subbands are related to the vertical, horizontal and diagonal searches respectively in the sense that a subband perpendicular to the search direction contributes more than other subbands.

4.5 Experiments

We evaluated several algorithms along with our algorithm using common settings as follows: CIF resolution, a fixed bitrate of 0.5 bpp and blocksize 16×16. The aim of the experiment was to determine, for each algorithm, the effectiveness in terms of PSNR against the efficiency in terms of computation. Computation was measured by counting the number of block difference operation.

RWSH[20], RWMH, RWMH with one-level decomposition (ONELEVEL), full-spatial search (SPATIAL), Three-step search (TSS), Diamond search (DS) and Hierarchical search (HIER) are compared and results are tabulated in Table. 1.

The RWSH system takes more time than RWMH due to the subsampling procedure to construct the tree structure from wavelet macroblocks (for more details, see [8]). The hierarchical method starts searching from the lowest level using only subbands at the current level and proceeds to the higher level with a reduced search window and more subbands included at corresponding level. The hierarchical search shows similar performance to RWMH but with 50% reduced computation. However, the hierarchical search requires five times as much computation as the full-spatial search. We also attempted a hierarchical search considering subbands at only the current level, motion vectors are easily mis-selected in the highpass due to the scattered representation. The RWMH with one-level decomposition shows comparable performance to the hierarchical search. Surprisingly, the TSS shows better performance than DS with comparable computation which contrasts the TSS and DS comparison found in other literature. As a motion-compensated redundant wavelet representation undergoes multi-phase to single-phase conversion[Eq(2)], it is not so transparent to measure matching. From another angle, one might consider block matching on high frequency filtered images as correlation. Empirically, we observe that optimal motion vectors found in the redundant wavelet domain are much closer to those found by correlation than those obtained by spatial block matching; a wavelet representation decomposes an input into sparse highpass and dense lowpass components, and weight factors between subbands can affect performance.

According to the result of Table. 1, our algorithm achieves consistently better performance than full-spatial search and with only 19% of the computation. PSNR deterioration is only 0.1dB compared with the TSS while computation is reduced by 56%. Our experiments show that utilizing perpendicular subbands to search direction is an effective technique to reduce computation, however the hierarchical structure remains difficult to use and futher work needs to be done.

4.6 Conclusion

We have discussed the role of the redundant wavelet in scalable video coding, the advantages of a phase-multi-hypothesis and its high computational requirement. We propose a fast motion search technique that utilizes directional subbands in combination with the three-step search. The technique reduces computation by 98%, 81% and 56% compared to RWMH, full-spatial search and the three-step search respectively, and yet still results in comparable performance to the three-step search method in the wavelet domain. Further improvement may be possible if the hierarchical structure can be utilized efficiently.

Chapter 5

Texture Synthesis

5.1 Texture Synthesis

The idea of texture Synthesis idea has been around for decades, and which first proposed by Garber[9] in 1981. A sample of texture reproduces an unlimited amount of image data which will be perceived by humans to be the same texture. This has been an active research area in computer graphics and is also known as image based texture synthesis. It has a strong influence in image based rendering. A sample of the real world is used to synthesize novel views rather than recreating the entire physical world from scratch. Texture synthesis can be categorised into two groups, parametric and non-parametric. The non-parametric group analyse the original sample of texture by its own rules and synthesize new texture whereas the parametric group finds a set of parameters to fit a certain model to the original texture data and reproduces texture using the parameters identified. However, being non-parametric, they lack an ability to analyse the characteristics present in a texture while the parametric texture synthesis can provide the characteristics through a set of parameter, which in turn could be utilised in segmentation and image recognition applications. Texture synthesis by affine transformation of a prototype block into another block, i.e. two-component analysis[11] and affine invariant image[3] falls in the parametric group. This offers promising performance for general texture synthesis as well as for compression applications. Later, a more fundamental approach by generative modeling of textons, which may be related to studies of optimal prototype selection, is mentioned.

5.2 Affine Transform based Texture Synthesis

In this section we refer to the work of Hsu, Calway, Wilson and Bhalerao [11, 3, 5]. The principal idea is analysed and implemented. We attempt to improve on the technique using a Gaussian mixture model and Expection maximisation and progress is briefly reported.

5.2.1 Preliminaries

Gaussian Mixture Model

A Gaussian Mixture Model (GMM)[4] is a type of probability density model which combines an arbitrary number of gaussian models. Each constituent Gaussian model is weighted by ω. The GMM can be expressed as

$$GMM(x) = \sum_{m}^{M} \Omega_m GM_m(x|\mu, \sigma) \tag{5.1}$$

where x, μ and σ is a vector of arbitrary dimension, a mean vector and the variance of the gaussian model. The gaussian model is defined as

$$GM(x|\mu, \sigma) = \frac{1}{2\pi^k|\Sigma_i|} exp(-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T) \tag{5.2}$$

where Σ is a covariance matrix.

Expectation Maximisation

Expectation Maximisation (EM)[4] is an estimation algorithm driven by the probability of membership to classes for each data point. The EM is a favourite in statistics due to its simplicity and fast convergence rate, but is apt to fall in local optima. Estimation is achieved by repeating two steps, E-step and M-step. An example follows with respect to the Gaussian mixture model. The E-step, computes expected classes for all data points for each class and given by

$$P(\omega_i|x_k, \mu_t, \Sigma_t) = \frac{p(x_k|\omega_i, \mu_i, \Sigma_i)\omega_i}{\Sigma_{j=1}^{c} p(x_k|\omega_j, \mu_j, \Sigma_j)\omega_j} \tag{5.3}$$

In the M-step, the maximum likelihood is estimated given new membership distribution for data

$$\mu_i = \frac{\Sigma_k P(\omega_i|x_k, \mu_t, \Sigma_t)x_k}{\Sigma_k P(\omega_i|x_k, \mu_t, \Sigma_t)} \tag{5.4}$$

$$\Sigma_i = \frac{\Sigma_k P(\omega_i|x_k, \mu_t, \Sigma_t)(x_k - \mu_i)(x_k - \mu_i)^T}{\Sigma_k P(\omega_i|x_k, \mu_t, \Sigma_t)} \tag{5.5}$$

$$\Omega_i = \frac{\Sigma_k P(\omega_i|x_k, \mu_t, \Sigma_t)}{numberofdata} \tag{5.6}$$

Levenberg Marquardt

The Levenberg Marquart (LM) method[22] is a non-linear data fitting algorithm and has become the standard of nonlinear least square routines. One can imagine this method as an extension of the simple Newton method. The important insight of this method is that the Hessian matrix can indicate how far the slope extends whereas the Jacobian matrix can tell only the gradient. A factor, λ is introduced to control the diagonal dominance of the Hessian matrix of second order partial derivatives of unknowns. Inceasing or decreasing the factor allows to move back and forth on the surface of least-squares. In other words, if the least-square increases we can change direction by the increasing factor, decreasing otherwise. This guarantees convergence unlike the Newton algorithm, however this method also is easily trapped in local minima. For more details, see [22].

5.2.2 Two-Component Analysis Method

Given a small prototype block, the entire image is approximated by small blocks exploiting affine invariance. The fourier transform is an essential part of this work due to the fact that the fourier transform separates an image into the magnitude spectrum exhibiting a linear feature of the image and the phase spectrum showing shift. This orthogonality is a very useful property as a linear transform can be estimated between two blocks without interfering with translation and vice versa. One can imagine how difficult it would be to find the optimal linear transform in the spatial domain without considering translation.
For the linear part, identifying two centroids in the magnitude spectrum that minimises the global variance reveals important linear features of the block. By aligning the centroids/vectors found in the prototype block and the target block, one can identify the affine matrix between two blocks easily. For translation, the position of maximum cross-

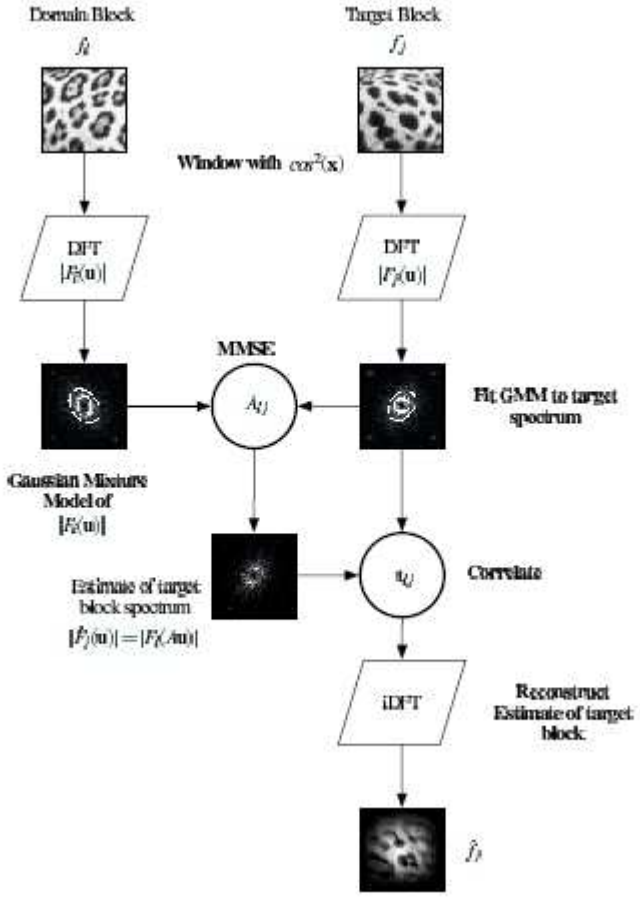


Figure 5.1: Affine Transform Estimation(taken from A. Bhalerao *et. al.* [3])

correlation represents the offset. The number of principal features often causes a problem. For instance, finding two centroids in the spectrum where only one exists and align with that of other block can produce stretched/shrinked approximation. This problem has been tackled in a different angle in [3] as illustrated in 5.1. Instead of searching for two centroids, the entire magnitude spectrum is modeled by Gaussian mixture model using LM, which in turn allows one to find affine matrix using LM again. Although the latter technique is computationally expensive, it is robust with respect to the number of strong linear features present in the spectrum and also is notable that it attempt to model the spectrum.

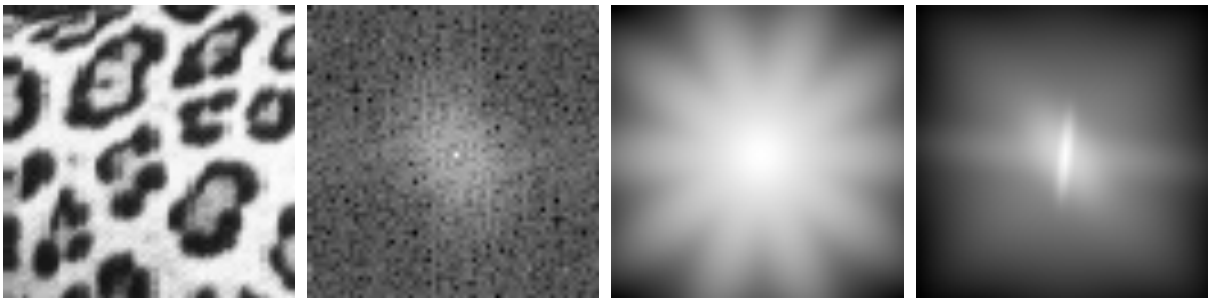
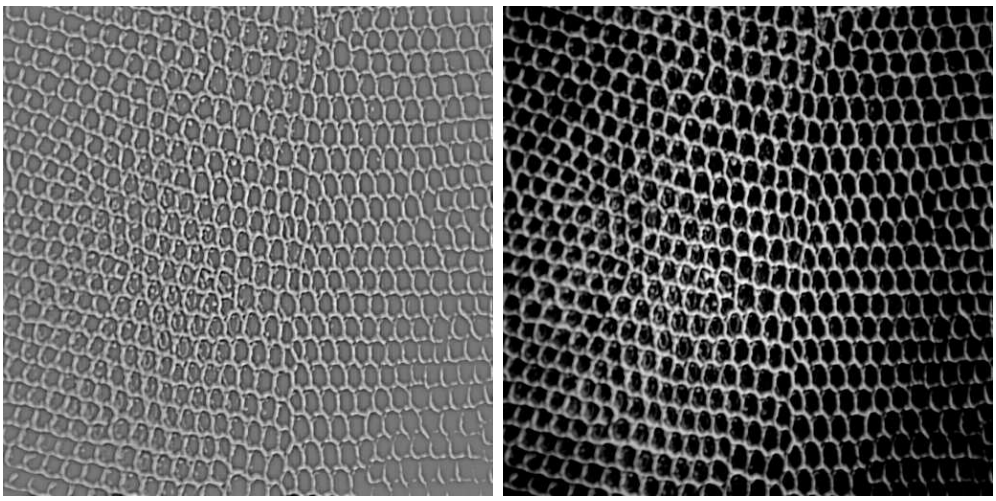


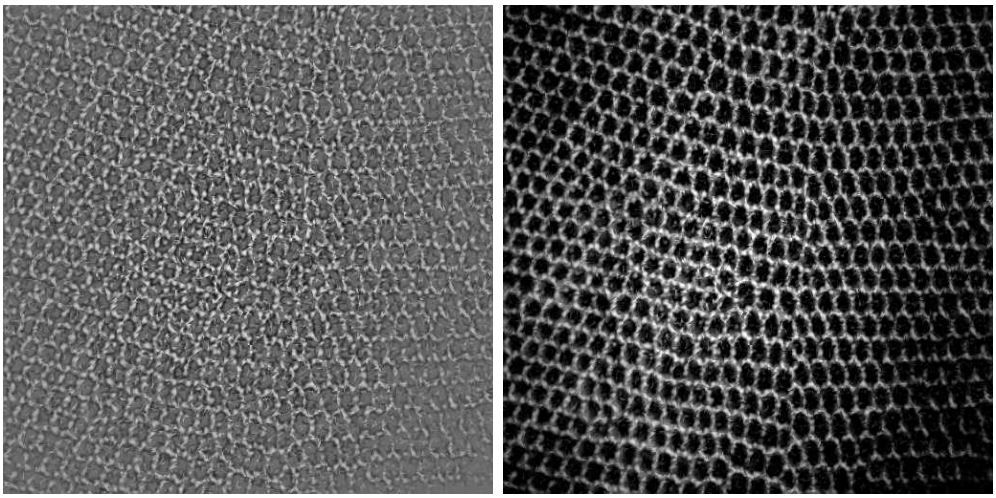
Figure 5.2: Magnitude modeling using EM : (left to right) image block, magnitude spectrum, gaussin mixture before EM iteration, and approximation after iteration

Alternative Strategy

A compromise between the two aforementioned methods is attempted by identifying the principle components in the magnitude spectrum using a gaussian mixture model and EM. The gaussian mixture model is adopted to model the magnitude spectrum in the hope that each constituent model can represent a different feature. Then, an affine matrix can be derived from them. First, the mixture is initialised as constituent gaussian models are evenly distributed between 0 and 360 degree. As shown in 5.2, a six-gaussian mixture model is used to approximate the magnitude spectrum while one component is fixed as non-directional and covering background. Second, a component is removed or fixed as background component after every two iteration whose degree of directionality (using ratio of eigenvalues) is less than a certain threshold. As it converges, components left *non-fixed* are assumed to represent different linear features, which in turn provides an affine matrix. This would be robust to the number of directional features present and computationally efficient. We have not obtained a descent result with this approach yet because gaussian models often overlap each other and also strongly directional ones with negligible weight mean little. As this work is still in progress, we need more empirical testing to determine the critical weight, the directionality threshold as well as a measure of similarity between components to be merged when obtaining the affine matrix. Another approach would be to incorporate a minimisation of inter-variance between components in EM but this is non-trivial considering the nature of the EM method.



Prototype Block- 



Prototype Block- 

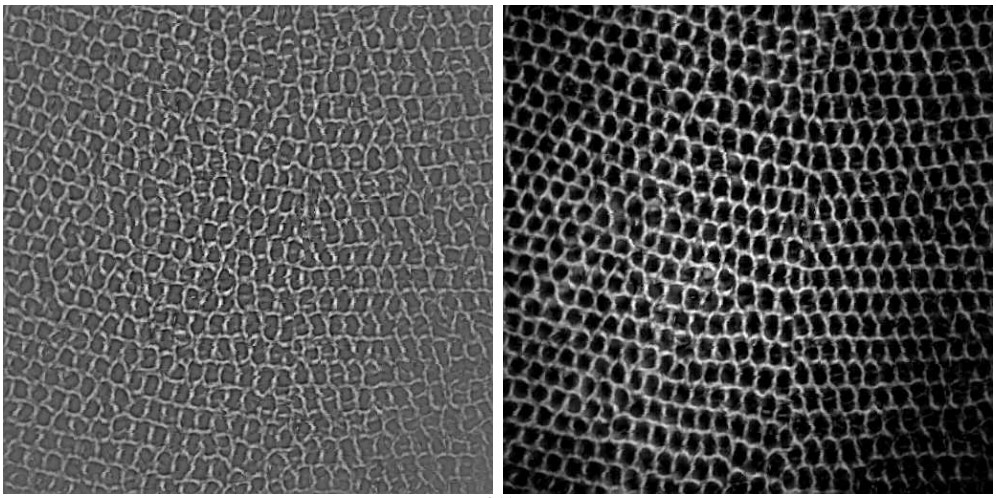
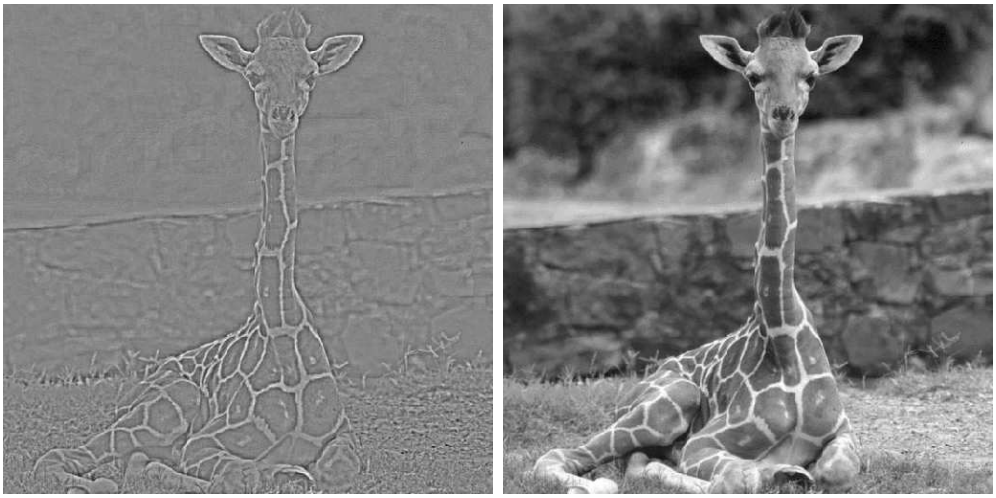


Figure 5.3: Synthesis with randomly chosen prototype : Reptile; (top)original (middle) synthesis using 16×16 block, (bottom) synthesis using 32×32

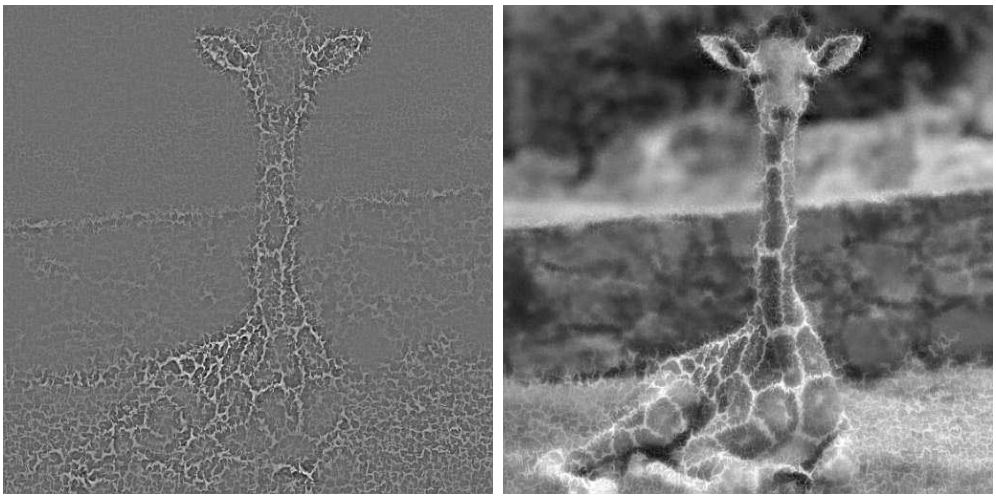
Prototype Block- 

Prototype Block- 

Figure 5.4: Synthesis with line prototype : Lena; (top)original (middle) synthesis using 16×16 block, (bottom) synthesis using 32×32



Prototype Block- 



Prototype Block- 

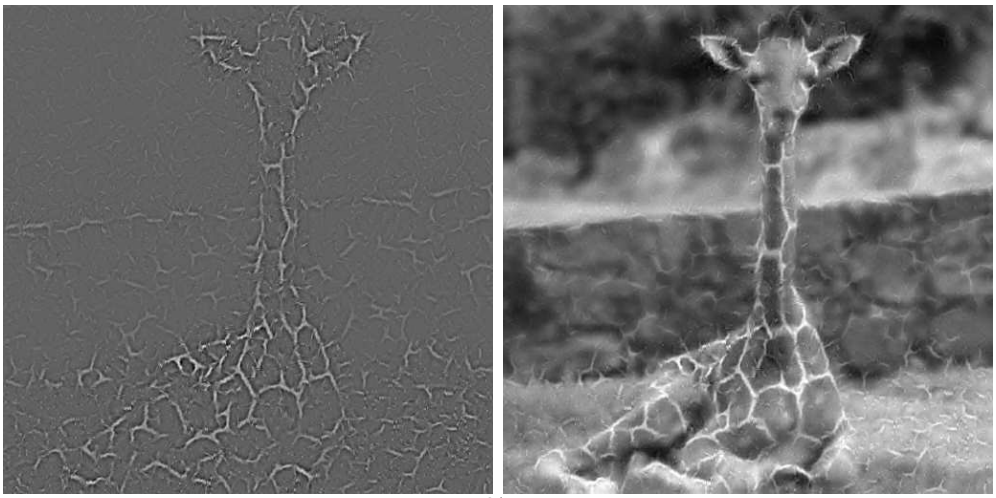
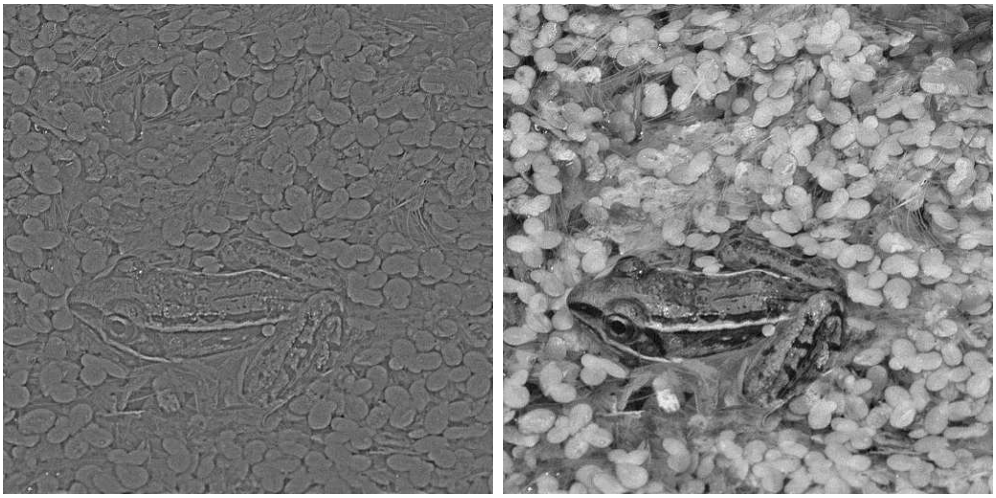
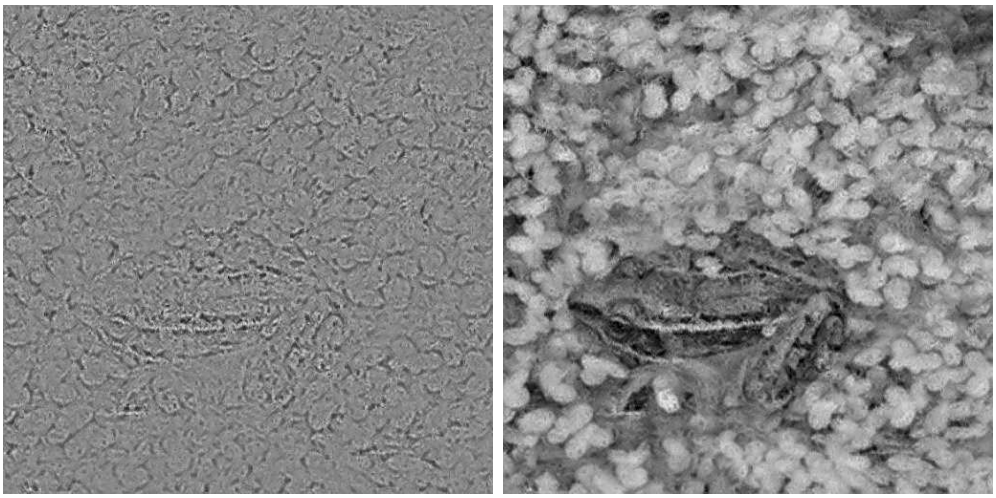


Figure 5.5: Synthesis with randomly chosen prototype : Giraffe; (top)original (middle) synthesis using 16×16 block, (bottom) synthesis using 32×32



Prototype Block- 



Prototype Block- 

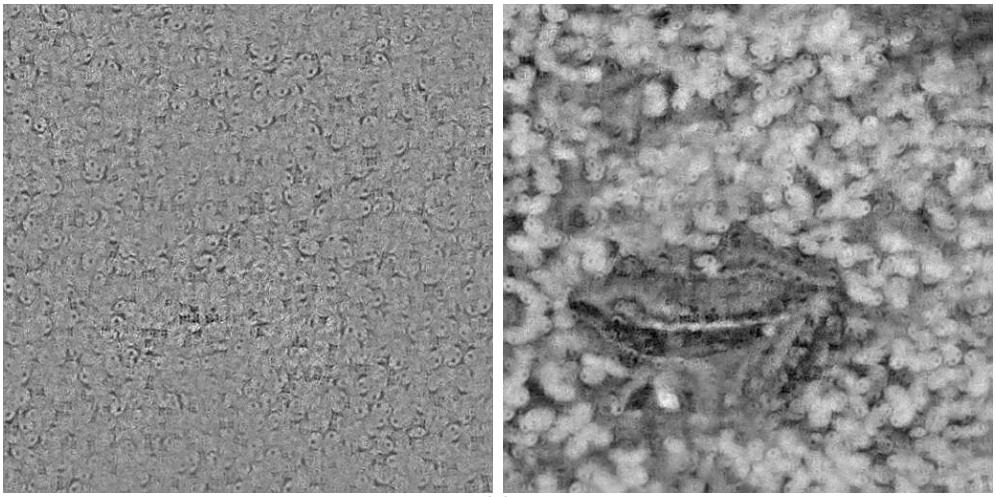
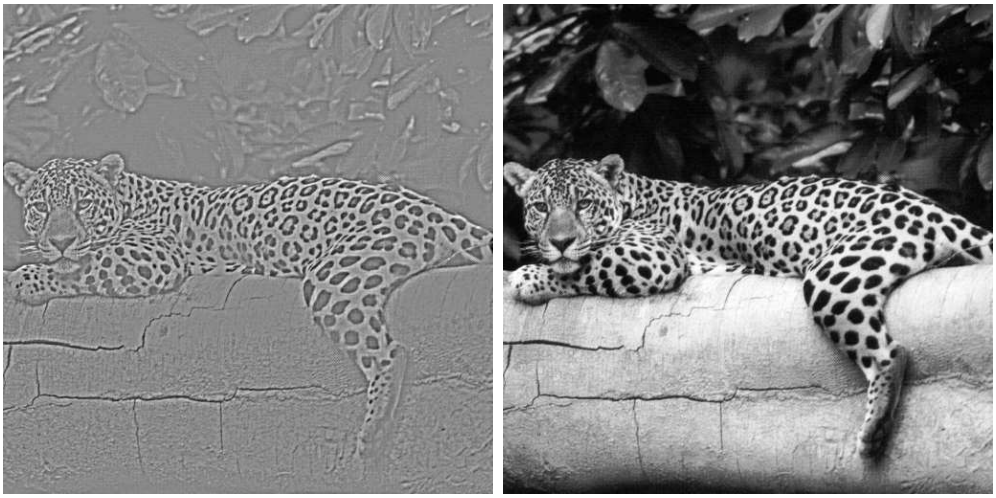
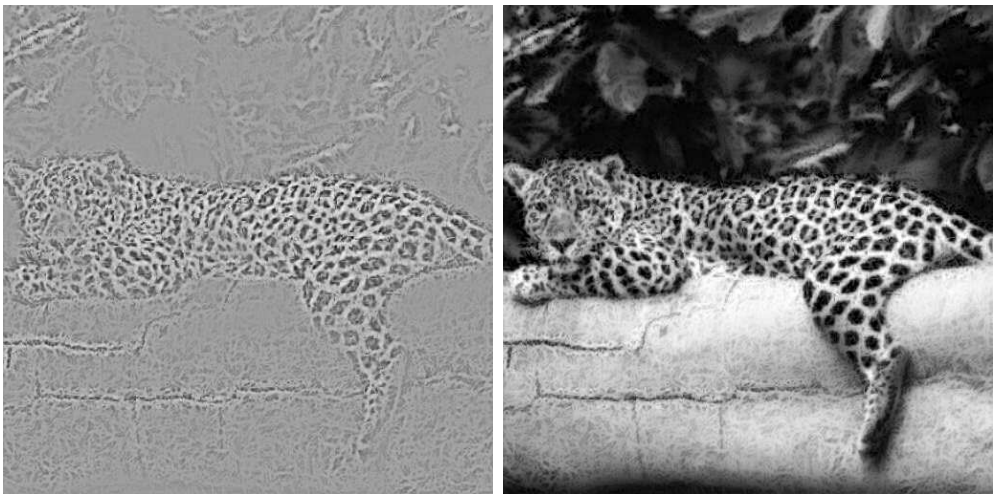


Figure 5.6: Synthesis with randomly chosen prototype : Frog; (top)original (middle) synthesis using 16×16 block, (bottom) synthesis using 32×32



Prototype Block- 



Prototype Block- 

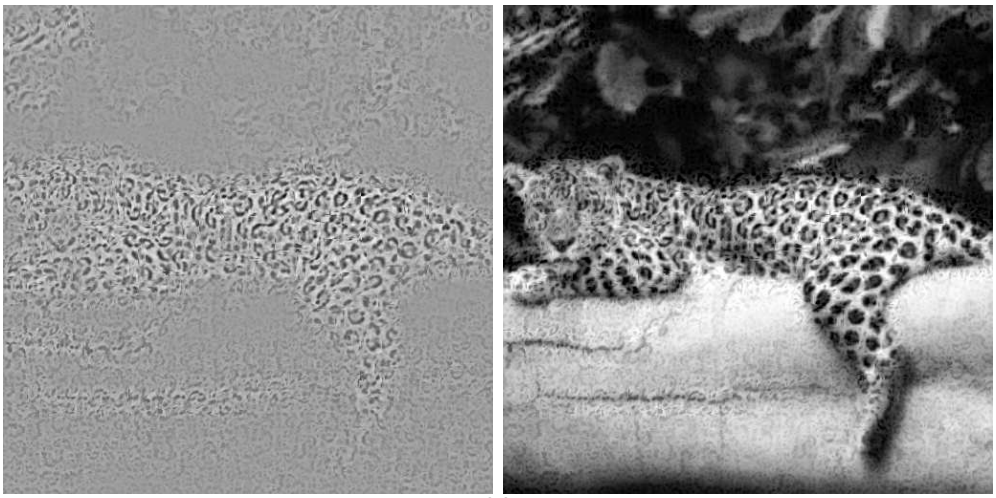


Figure 5.7: Synthesis with randomly chosen prototype : Jaguar; (top)original (middle) synthesis using 16×16 block, (bottom) synthesis using 32×32

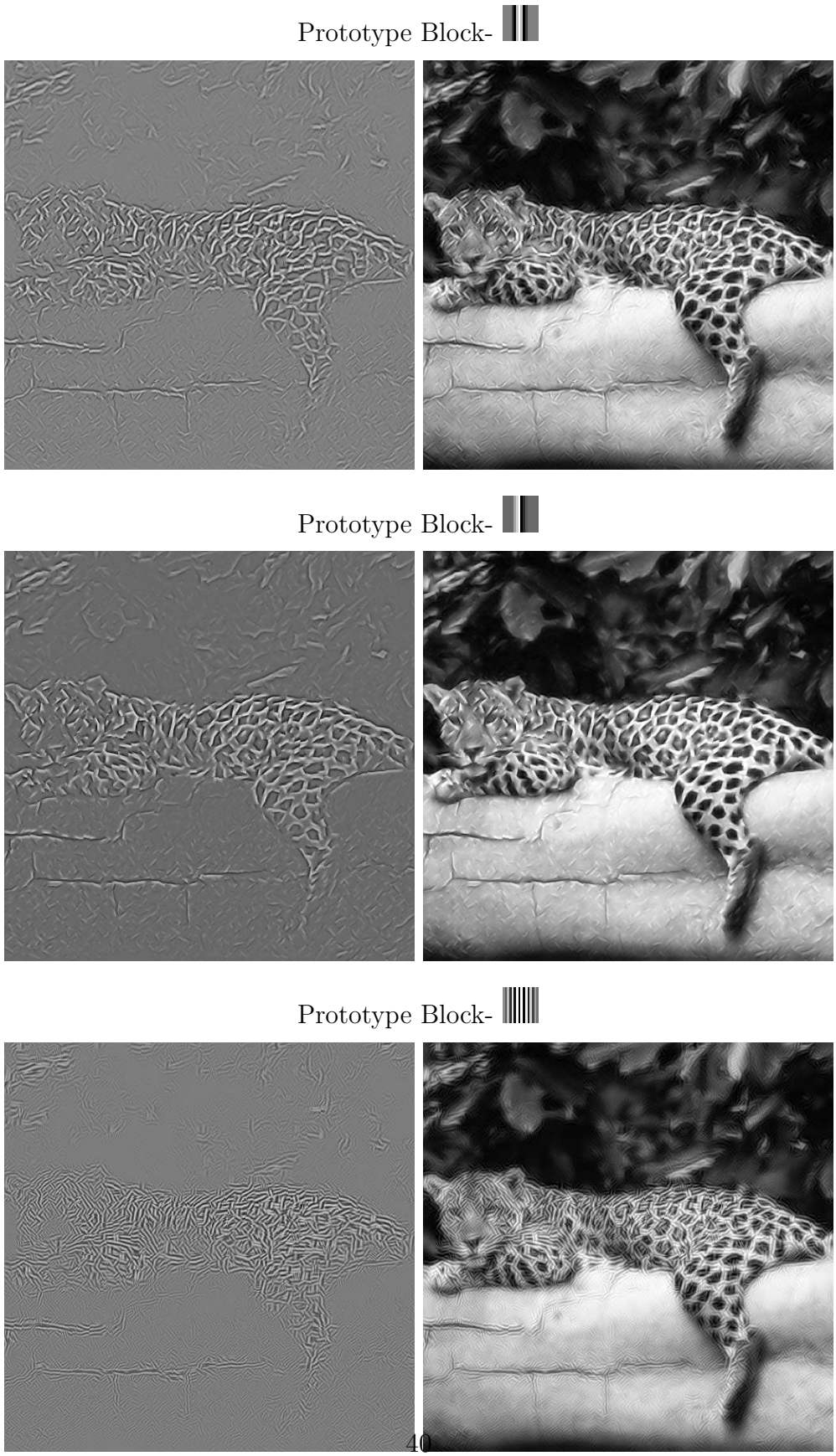


Figure 5.8: Using various prototypes; (top)line (middle) cliff, (bottom) ridge

Example of Texture Synthesis

The two-component analysis algorithm is replicated and tested for several images as shown in Fig.5.3∼Fig.5.8. The work above has well-established the framework for transforming one block to another. However, one fundamental problem remains to be solved, that is finding the optimal prototype. The reconstruction quality greatly depends on the selected prototype block. In [3], the problem is tackled by multiple prototype blocks identified using the K-means clustering algorithm and an improved approximation is shown. In the next section, we pursue the texton that is the fundamental and minimal structure of texture.

5.3 Generative Model based Texture Synthesis

5.3.1 Independent Component Analysis

Independent Component Analysis (ICA) is a technique that recovers a set of independent data from a set of observed data. It is assumed that each observed data is a linear combination of each of the independent data, and that there are an equal number of observed data and independent data. A common example is the *Cocktail party problem*. If you imagine that two people are speaking at the same time in the party and each has microphone recording conversation. This can be expressed as

$$obs_1 = a_{11}v_1 + a_{12}v_2 \tag{5.7}$$

$$obs_2 = a_{21}v_1 + a_{22}v_2 \tag{5.8}$$

where obs_1, obs_2, v_1 and v_2 are the observed data and voice respectively, and a_{11}, a_{12}, a_{21} and $a22$ are mixing parameters. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components. The independent components are latent variables, not directly observable. The goal is to find the mixing matrix and the original data. In this context, The ICA is very closely related to the method called blind source separation[1]. The ICA starts from an assumption that each source are independent of each other. Independence implies a distribution of one source is uniform regardless of the other source. We shall see why independence is required. First, the above equations are re-written as

$$Obs = Mix \cdot V \tag{5.9}$$

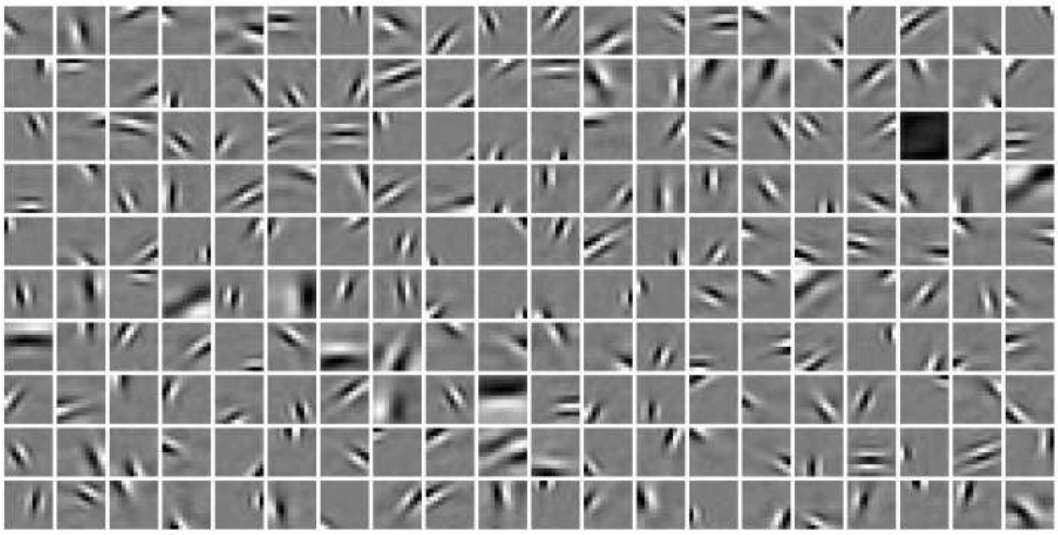


Figure 5.9: Image bases learned with sparse coding (taken from [23])

Then, the original data can be recovered by identifying the mixing matrix, Mix as follows.

$$V = Mix^{-1} \cdot Obs \tag{5.10}$$

Thus, finding the mixing matrix, Mix is a key. Identifying the mixing matrix is possible if and only if the source is non-gaussian,independent or non-symmetric as mentioned. Independent relation of sources can be expressed as

$$P(A, B) = P(A)P(B) \tag{5.11}$$

$$P(A|B) = P(A) \tag{5.12}$$

where A and B are the original data. This is important with respect to 5.9 and 5.10 as Mix can be obtained by assuming V has a joint uniform distribution. In turn, this implies the observed data directly shows deformation of the joint uniform distribution. Then the vector defining an area of the joint distribution corresponds to the column vectors of the mixing matrix. Although this is not a practical solution, it shows insight into the ICA. In practice, a mixing matrix is rather estimated by maximising the non-gaussian nature using a certain measure, i.e. kurtosis or negentropy. More details can be found in [1].

5.3.2 Texton

Up until today, the texton remains an ambiguous concept and whose mathematical model has not been defined. However, it is certain that the texton is a basic pre-attentive structure to human visual perception. In this context, important work has been carried out by Olshausen and Field[18]. Over-complete image bases from natural image patches with the sparse coding idea is learned assuming the patch is a generative model of the bases. Fig.5.9 shows those bases. Unlike the orthogonal and complete bases in the Fourier or Wavelet transform, the resulting bases are highly correlated. The bases are clearly well localised both in frequency and in space. Sparse coding is that if one of the bases contributes much to the linear combination that makes up the original data, all others will not contribute or only to a very small degree. The hidden idea on which the sparse coding is based is the ICA. The ICA decomposes images as a linear superposition of a set of image basis which minimises some measure of dependence between the coefficients of these bases[2]. The resulting bases can be imagined as being neurons down the visual pathway that get excited independent of each other when faced with visual stimuli.

5.3.3 Generative Model for Texton

In [29], the aforementioned texton idea is extended to model natural images as well as a sequence of images. A three-level generative image model is introduced assuming that an image is a superposition of a number of image bases selected from an over-complete dictionary at various locations, scales, and orientations. The set of bases are generated by a smaller number of texton elements. The generative model is further extended to motion sequence and lighting variations with the geometric, dynamic and photometric structures of the texton representation.

The geometric structure explains spatial deformation of a texton. The dynamic structure is addressed by characterising the motion of a texton with a Markov chain model. The photometric structure represents a texton being a three-dimensional surface under varying illuminations. The work has shown great potential in synthesising not only a block of an image but also deformations through time with textons as primitives. More details and examples can be found in [29].

Chapter 6

Conclusion

6.1 Wavelet based Video Coding

Wavelet based video coding provides scalability to the compression of video data over time. In particular, a motion compensated temporal filtering scheme provides substantial improvements over 3D-wavelet coding without motion compensation. The potential use of mesh-based motion compensation is discussed and a method to re-generate a motion adaptive mesh without overhead information is reported. Redundant wavelet based coding can provide better performance using phase-diversity but requires high computation during motion estimation. Fast motion estimation using selective consideration of the subband perpendicular to the search direction is proposed. The redundant wavelet can also be advantageous when re-generating a mesh in the decoder. However, it is not an easy task to deploy it in the wavelet coder without transmission of overhead information due to inaccessibility of the exact same mesh used in the encoder.

6.2 Video coding via Texture Synthesis

Affine transform-based texture synthesis has given a realistic perspective to its implementation in compression applications given the optimal prototype. However, finding the optimal prototype is not a trivial task. The notable framework extracting over-complete bases from a natural image using independent component analysis and sparse coding seems important in connection with finding an optimal prototype, in the sense that it can decompose both prototype and target into a set of texture primitives. Further extension using over-complete bases as a generative model of the texton is a wholly different approach to

texture synthesis.

6.3 Conclusion and Future direction

The two work items conducted this year seem little related. However, we realise that the affine transform is a common core factor considering mesh-deformation, block-based texture synthesis, and even in ICA. In the mesh case, it is straight-forward use of the affine transform to warp a triangle in the reference frame to a different shape of triangle in the anchor frame, nonetheless it has many advantages over block-based compensation. In texture synthesis, the symmetrical redundancy is exploited using an affine transform that the conventional compression systems have not attempted. Lastly, ICA is about estimating deformation of uniformly distributed data, which can be expressed by an affine transform. It would be interesting to see how the affine transform can be further combined with the theories and approaches of video compression. Currently, an efficient method of compressing the affine parameters resulted from texture synthesis is being sought.

For future research, we plan to explore following. Texture synthesis based on an affine transform has shown promising performance for still images in terms of visual quality. For video, a direct extension to three-dimensions would be using a cube instead of a block with an image sequence partitioned to be of the same size. A cube to cube transform would require the estimation of three centroids and having to deal with a 4×4 homogeneous matrix. Although it looks straight-forward, we are reminded that it is likely to offer less-pleasing results as we use a more limited form of prototype by restricting time as well. However, it is worthy of investigation.

Another interesting approach would be using the generative model. As seen in the over-complete bases, there is quite symmetrical redundancy between the bases. This would allow us to find a reduced set of bases by exploiting affine invariance. The resulting bases could be assumed as general bases that can produce any block with a linear combination of the bases. In turn, the generated block could serve as a prototype for a certain class of texture.

Bibliography

[1] E. Oja A. Hyva"rinen, J. Karhunen. *Independent Component Analysis.* Wiley and Sons, 2001.

[2] A.J. Bell and T.J. Sejnowski. An information maximization approach to blicd separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, 1995.

[3] A. Bhalerao and R. Wilson. Affine invariant image segmentation. In *BMVC.*, Kingston University, UK, 2004.

[4] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[5] A. D. Calway. Image representation based on the affine symmetry group. In *IEEE ICIP.*, pages 189–192, Sep. 1996.

[6] S.J. Choi and J.W. Woods. Motion-compensated 3-d subband coding of video. *IEEE Trans. Image Process.*, 8:155–167, 1999.

[7] S. Cui, Y. Want, and J.E. Fowler. Mesh-based motion estimation and compensation in the wavelet domain using a redundant transform. In *IEEE ICIP.*, volume 1, pages 693–696, Montreal Canada, Sep. 2002.

[8] S. Cui, Y. Want, and J.E. Fowler. Multihypothesis motion compensation in the redundant wavelet domain. In *IEEE ICIP.*, volume 2, pages 53–56, Barcelona Spain, Sep. 2003.

[9] David Donovan Garber. *Computational Models for Texture Anlaysis and Texture Synthesis.* PhD thesis, University of Southern California,Department of Electrical Engineering, Signal & Image Processing Institute, 1981.

[10] S.-T. Hsiang and J.W. Woods. Embedded video coding using invertible motion compensated 3-d subbnad/wavelet filter bank. *Signal Process.: Image Comm.*, 16:705–724, 2001.

[11] T.I. Hsu and R. Wilson. A two-component model of texture for analysis and synthesis. *IEEE Trans. Image Process.*, 7(10):1466–1476, 1998.

[12] C. Huang and C. Hsu. A new motion compensation method of image sequence coding using hierarchical grid interpolation. *IEEE Trans. CSVT.*, 4(1):42–51, Feb. 1994.

[13] ISO/IEC 15444-1: JPEG2000 Image Coding System: Core Coding System ISO/IEC JTC1/SC29/WG1, 2000.

[14] G. Karlsson and M. Vetterli. Three-dimensional subband codin gof video. In *IEEE ICASSP.*, pages 1100–1103, New York, Apr. 1988.

[15] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro. Motion-compensated interframe coding for video conferencing. *IEEE NTC*, pages 531–534, 1981.

[16] Y. Nakaya and H. Harashima. Motion compensation based on spatial transformations. *IEEE Trans. CSVT.*, 4(3):339–356, Jun. 1994.

[17] J. Ohm, M. Schaar, and J. Woods. Interframe wavelet coding-motion picture representation for universal scalability. *Signal Process.: Image Commun.*, 19(9):877–908, Oct. 2004.

[18] B.A. Olshausen and D.J. Field. Sparce coding with an over-complete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

[19] H. Park, G.R.Martin, and A.C. Yu. Fast motion estimation based on the redundant wavelet for 2d+t wavelet coder. In *VLBV.*, Sardinia, Italy, Sep 2005. to appear.

[20] H. Park and H. Kim. Motion estimation using low-band-shift method for wavelet-based moving-picture coding. *IEEE Trans. Image Process.*, 9:577–587, Apr. 2000.

[21] H. Park, A.C. Yu, and G.R. Martin. Progressive mesh-based motion estimation using partial refinement. In *VLBV.*, Sardinia, Italy, Sep 2005. to appear.

[22] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing.* Cambridge University Press, 2000.

[23] R. P. N. Rao, B. A. Olshausen, and M. S. Lewicki. *Probabilistic Models of the Brain: Perception and Neural Function*. MIT Press, 2002.

[24] A. Secker and D. Taubman. Motion-compensated highly scalable video compression using an adaptive 3d wavelet transform based on lifting. In *IEEE ICIP.*, pages 1029–1032, Thessaloniki, Greece, Oct. 2001.

[25] A. Secker and D. Taubman. Highly scalable video compression using a lifting-based 3d wavelet transform with deformable mesh motion compensation. In *IEEE ICIP.*, volume 3, pages 749–752, Montreal Canada, Sep. 2002.

[26] A. Secker and D. Taubman. Lifting-based invertible motion adaptive transform(limat) framework for highly scalable video compression. *IEEE Trans. Image Process.*, 12:1530–1542, Dec. 2003.

[27] C. Toklu, A. Erdem, M. Sezan, and A. Tekalp. Traking motion and intensity variations using hierarchical 2-d mesh modelling for synthetic object transfiguration. *Graphical Models and Image Process.*, 58(6):553–573, Nov. 1996.

[28] J.C. Ye and M. van der Schaar. 3-d lifting structure for sub-pixel accuracy motion compensated temporal filtering in overcomplete wavelet domain. *MPEG2003/M9554, ISO/IEC JTC1/SC29/WG11.*

[29] .C. Zhu, Y. Z. Wang C. Guo, and Z. J. Xu. What are textons? *IJCV. Special Issue on Texture*, 2005.